

*Application
for
United States Patent*

To all whom it may concern:

*Be it known that, Jens Haulund and Mark Lutian
have invented certain new and useful improvements in*

***AN APPARATUS AND METHOD FOR PROVIDING DOMAIN NAME SERVICES TO
MAINFRAME RESOURCE MAPPING***

of which the following is a full, clear and exact description:

**AN APPARATUS AND METHOD FOR PROVIDING DOMAIN NAME
SERVICES TO MAINFRAME RESOURCE MAPPING**

PRIORITY

5 [0001] This application claims priority to the provisional U.S. patent application entitled, DYNAMIC RESOURCE MAPPING IN A TCP/IP ENVIRONMENT, filed December 27, 2000, having a serial number 60/258,437, the disclosure of which is hereby incorporated by reference.

10 **FIELD OF THE INVENTION**

[0002] The present invention relates generally to domain name system mapping. More particularly, the present invention relates to domain name system resource mapping in a mainframe environment.

15 **BACKGROUND OF THE INVENTION**

[0003] The Domain Name System (DNS) enables a user or process to ask for a host (computer) by name without having to know the Internet Protocol (IP) address of the host. A common example of DNS is the IP address for www.inrange.com that results in the following sequence of two IP datagrams:

20 DNS_name_query www.inrange.com is sent to a DNS server as configured in the IP host's gateway statement.

25 DNS_name_response www.inrange.com is on IP address 192.1.1.1 is the reply from the DNS server.

[0004] The current well-defined and understood concepts of DNS are found in *RFC 1035 Domain Names-Implementation and Specification*, P. Mockapetris, November, 1987, which is further enhanced in several additional *RFCs, particularly RFC 2136 Dynamic Updates in the Domain System (DNS UPDATE)*, P. Vixie, *et al.*

[0005] FIG.1 illustrates the basic implementation of DNS Protocol logic. Using the DNS protocol described above, a user/DNS requester 10 is able to ask for a host by computer name rather than knowing the IP address of this host computer. As an example, the user/DNS requester 10 requests a host selected 10 from among various IP hosts 14, 16, 18, 20 on an IP network having an IP network name CUSTOMER.COM. The user/DNS requester 10 sends a message to the DNS server 12 with a DNS request for an IP address corresponding to the name of the requested host. The DNS server 12 maintains a table of hostnames as well as one or more IP addresses associated with each hostname. The DNS 15 server 12 receives the request from the user/requester 10, and using the table of host names, resolves the hostname to a corresponding IP address. The DNS server 12 then sends a reply to the user/requester 10 containing the IP address.

[0006] In the present example, a hostname lookup of “hostname.customer.com” returns with the IP address 192.1.1.1. A hostname 20 lookup of allhosts.customer.com returns one of four listed IP addresses.

[0007] The reason that more than one IP address is associated with a given hostname is to allow for multiple physical processors to deal with incoming requests to the hostname. Thus, the DNS server 12 responds to a request for hostname customer.com with any one of the IP hosts 14,16,18, 20. With the DNS 25 logic described, a simple round-robin assignment of a physical processor takes place.

[0008] Currently, there is a need in the mainframe environment for DNS capability. DNS has not been implemented into the mainframe environment due to the inability of mainframes to mesh coherently with other mainframes and open computers. However, “locked” within these mainframes are a number of resources that contain important information. These resources, as well as information contained therein, is not accessible without knowing the actual native standards of the mainframe world. Accordingly, it is desirable to provide a system that provides DNS capability to resources in a mainframe environment.

10

SUMMARY OF THE INVENTION

[0009] In a first aspect of the invention, a system is provided for DNS mapping to resources in a mainframe environment. The system includes a locally managed DNS server and a DNS protocol to allow a client to request a mainframe resource. In response, the DNS server returns the selected address of the client.

15

[0010] In another aspect of the invention, a client side DNS process is provided for collecting resource performance characteristics. In response to a request for a mainframe resource, the DNS server returns the selected address of the client based upon collected machine performance characteristics of at least one client.

20

[0011] In another aspect of the invention, the system includes mainframe resource polling. This is accomplished by the DNS server either polling the resource and requesting information or having the resource transmit its operability status.

25

[0012] In a further aspect of the system, the DNS server removes the resource for accessing requests. As a result, the resource becomes unavailable through the DNS server. The resource is removed for requests when the resource

does not respond to a status or operability inquiry. If, after a period of time, the resource begins to retransmit operability status or responds to inquiries from the DNS server, then the resource is placed back into the DNS server for accessibility.

5 [0013] In an alternate embodiment of the invention, a method for DNS mapping in a mainframe environment is provided. The steps of the method include receiving a DNS request for a resource in a mainframe environment, selecting the resource from among registered mainframe resources and providing an address corresponding to the mainframe resource.

10 [0014] In a further aspect of the invention, the step of collecting performance characteristics on mainframe resources is provided. From these characteristics, an analysis is performed to determine what mainframe resource to select upon a request to the DNS server.

15 [0015] In a further aspect of the alternate embodiment, the step of polling the resource is provided to ensure its operability. In conjunction with the polling, the step of disassociating a resource from the mainframe DNS mapping is provided. This step is taken in response to non-transmittal of the operability status of the resource.

20 [0016] In a further embodiment of the present invention, a system for DNS mapping in a mainframe environment includes means for receiving a DNS request for a resource in a mainframe environment, means for selecting the resource from among registered mainframe resources and means for providing an address corresponding to the mainframe resource.

25 [0017] In another aspect of the alternate embodiment, the system includes means for collecting performance characteristics on mainframe resources. In the preferred embodiment, the means for selecting the resource chooses a suitable

resource for access through an analysis of similarly requested mainframe resources.

[0018] In another aspect of the preferred embodiment, means for polling the resource to ensure its operability is provided. This element ensures that the
5 resource is operable when a request for the resource is received.

[0019] There has thus been outlined, rather broadly, the more important features of the invention in order that the detailed description thereof that follows may be better understood, and in order that the present contribution to the art may be better appreciated. There are, of course, additional features of the invention
10 that will be described below and which will form the subject matter of the claims appended hereto.

[0020] In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and to the arrangements of the
15 components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein, as well as the abstract, are for the purpose of description and should not be regarded as limiting.

[0021] As such, those skilled in the art will appreciate that the conception upon which this disclosure is based may readily be utilized as a basis for the designing of other structures, methods and systems for carrying out the several purposes of the present invention. It is important, therefore, that the claims be regarded as including such equivalent constructions insofar as they do not depart
25 from the spirit and scope of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] FIG. 1 is a block diagram illustrating the concept of DNS protocol.

[0023] FIG. 2 is a block diagram of a typical mainframe environment.

[0024] FIG.3 is a tree diagram of the DNS system as related to standard
5 TCP/IP protocol and the present invention

[0025] FIG.4 is a block diagram of the present invention in a mainframe
environment.

DETAILED DESCRIPTION OF PREFERRED

10

EMBODIMENTS OF THE INVENTION

[0026] FIG.2 is a block diagram of a mainframe environment. The environment is connected through an ESCON director 22. The director 22, a dynamically modifiable switch, interconnects the mainframe computers 24,26 with each other and with attached devices 28a, 30a, 32a, 34a using optical fiber technology. Within each mainframe 24, 26 and devices 28a, 30a, 32a, 15 34a, there are potential applications or information that can be presented through the DNS.

15

[0027] In the preferred embodiment, INRANGE drivers 28b, 30b, 32b, 20 34b are front loaded at the devices 28a, 30a, 32a, 34a. The location of the driver in the dataflow gives the drivers 28b, 30b, 32b, 34b access to all information being transmitted across the channels 36, 38, 40, 42 of the ESCON director 22. The driver makes decisions based upon label information it receives in the mount request. The driver can be configured to interpret information at given offsets in the data and pass relevant information to the 25 DNS.

[0028] FIG.3 is a tree diagram of the DNS system as related to standard TCP/IP protocol and the present invention. The top or root level 44 domain is managed by the “administrators” of the Internet. From the root level 44, the Internet is divided into subdomains of which the consuming public is more aware such as .COM 46, .EDU 48 and .GOV 50. These domains are then broken into more specific domain such as ACME.COM 52, DELTA-AIR.COM 54 and WHITEHOUSE.GOV 56. The invention takes advantage of this division of the domains because each level below the root domain becomes more and more locally managed, which provides decentralized administration.

10

[0029] For example, if ACME.COM 52 has their domain under the .COM level, then everything under ACME.COM is managed by the Acme Company. With the present invention, a Message Director System (MD9000) made by Inrange Company of Lumberton, New Jersey that is part of Acme’s mainframe environment inherits a label in the DNS world such as MD9000.ACME.COM. In the configuration of the Acme’s DNS system, the administration of the subdomain MD9000.ACME.COM is delegated from ACME.COM DNS to MD9000.ACME.COM.

15

[0030] A DNS request for the hostname RESJOB.MD9000.ACME. 20 CUSTOMER returns the Internet Protocol (IP) address of the *resjob* 60 host. The same procedure is followed with *someotherjob* 62 and *device571* 64.

20

[0031] In a mainframe environment, which incorporates the MD9000 product by INRANGE, it is possible that the three hosts *resjob* 60, *someotherjob* 62 and *device571* 64 appear to exist in the MD9000 subdomain 25 58. In reality, only the names exist. The hosts exist virtually inside one or more MD9000 machines.

[0032] FIG.4 is a block diagram of the present invention as employed in a mainframe environment. The ESCON director 22 is connected to devices 28a, 30a, 32a, 34a in a mainframe environment. In the preferred embodiment, the devices 28a, 30a, 32a, 34a initiate contact with a DNS server 66. This is 5 done to alert the DNS server 66 as to its availability and thus making it available to others that desire access to the information by requesting it through a DNS address instead of the specific numerical address of the computer (e.g., www.frequentflier.com).

[0033] The preferred embodiment utilizes a generic uni-directional 10 communications method to accomplish an availability transmission 68 from device 28a to DNS server 66. However, other methods such as bi-directional or uni-directional transmission from the server 66 to the device are within the scope and spirit of this invention.

[0034] In an example of the preferred embodiment, a tape is mounted 15 at the address RES1.TEST.PNRDUMP on device address 571, which is on LPAR RES1. The names RES1.TEST.PNRDUMP and LPAR1.571 are transmitted from the MD9000 to the DNS server and then become available for lookup.

[0035] If a host on the Internet or Intranet issues the command: 20
“*ping res1.test.pnrdump.md9000.acme.com*”

The IP address 192.1.2.3 is returned by the DNS server lookup. Similarly, the command “*ftp lpar.571.md9000.acme.com*” returns the IP address 192.1.2.3.

[0036] In a traditional DNS environment, load balancing is achieved 25 using a round-robin algorithm in the DNS configuration file. The following is an example of this method:

	goodserver.md9000.acme.com	192.1.2.1 // First Server
	goodserver.md9000.acme.com	192.1.2.1 // Second Server
	goodserver.md9000.acme.com	192.1.2.1 // Third Server
	goodserver.md9000.acme.com	192.1.2.1 // Fourth Server
5	goodserver.md9000.acme.com	192.1.2.1 // Fifth Server

[0037] The first request from an IP host to lookup “*goodserver*” is given the address 192.1.2.1. The requests following the initial request are given 192.1.2.2 and 192.1.2.3 and so on. For this concept to function correctly, the 10 DNS cache maintained by IP hosts in the network should have very short timeouts and the Time-to-Live value returned in the DNS response be set to zero.

[0038] In the present invention, the round-robin algorithm can and does 15 exist for commonly named resources. For example, if the job TEST.PNRDUMP is run on different hosts simultaneously, the hosts in the IP environment can request TEST.PNRDUMP.MD9000.ACME.COM. From this request, they are directed to one of many MD9000 processors. This scenario is especially usefully 20 in situations where the workload is distributed to a high number of IP hosts for processing. The following is an illustration of this process.

	test.pnrdump.md9000.acme.com	192.1.2.1	//First MD9000
20	test.pnrdump.md9000.acme.com	192.1.2.2	//Second MD9000
	test.pnrdump.md9000.acme.com	192.1.2.3	//Third MD9000
	test.pnrdump.md9000.acme.com	192.1.2.4	//Fourth MD9000
	test.pnrdump.md9000.acme.com	192.1.2.5	//Fifth MD9000
	test.pnrdump.md9000.acme.com	192.1.2.6	//Sixth MD9000

25 [0039] For specific requests, the request is prefixed with the mainframe name (e.g., RES1). From this request, the following entries in the table are used:

	res1.test.pnrdump.md9000.acme.com	192.1.2.1	//First MD9000
30	res2.test.pnrdump.md9000.acme.com	192.1.2.2	//Second MD9000
	res3.test.pnrdump.md9000.acme.com	192.1.2.3	//Third MD9000
	res4.test.pnrdump.md9000.acme.com	192.1.2.4	//Fourth MD9000

res5.test.pnrdump.md9000.acme.com 192.1.2.5 //Fifth MD9000
res6.test.pnrdump.md9000.acme.com 192.1.2.6 //Sixth MD9000

[0040] In a more specific example, the airline carriers' computer system comprises a number of mainframe technologies of which traditional or standard 5 DNS does not support. The information on these systems is essential for many aspects of their business. Contained in these systems are such things as frequent flyer data, special meal data and fare code data.

[0041] Access to this type of data in a mainframe environment was difficult especially for other mainframes and open computer systems. With the 10 Inrange MD9000, access to this data has become easier without tremendous code writing in languages for which programmers are hard to find. As a result of the MD9000, this data once deemed accessible to only a few is now available to those who request access.

[0042] With the present invention, whether it is in an Intranet or Internet 15 environment, the requester does not need to remember the numerical address of the information. Therefore, if the requester requests access to the frequent flyer information on the internal mainframe system, all they have to do is send out a request for FREQUENTFLYER.ACME.COM as opposed to 192.2.1.

[0043] When the request for FREQUENTFLYER.ACME.COM is sent, 20 the DNS server 66 receives the request and proceeds to analyze its table of names for the corresponding IP address. If the resource FREQUENTFLYER.ACME.COM is associated with more than one IP address, the DNS server 66 determines from the collected performance characteristics what resource is most capable of handling it. Furthermore, if the resource FREQUENTFLYER.ACME.COM has become unavailable and fails to poll the DNS server 66, then 25 the FREQUENTFLYER.ACME.COM becomes unavailable for DNS lookup.

[0044] The preferred embodiment is dynamic in nature in that the DNS server directs the request to the resource most capable of handling it. As discussed previously, this is based on the collection of performance characteristics by the DNS server 66 with respect to the various resources. The 5 DNS server 66 analyzes performance information such as resource accessibility, resource processing time and number of requests prior to transmitting an IP address. The analysis determines the resource most able to handle the request.

[0045] The preferred embodiment also is dynamic in nature in that the DNS server 66 is privy to the operability of a resource. When a request arrives 10 at the DNS server 66, the DNS Server 66 is knowledgeable of the fact as to accessibility of the resource. Unlike standard DNS, the request is not returned with the IP address if the resource is listed as non-operable. Standard DNS does not determine the operability of a resource when returning an IP address

[0046] The DNS server 66 is aware of the resource operability by a 15 method known as polling. For example, the DNS server 66 polls the resource over a period of time. The resource transmits a response to the server making it aware of its operability status.

[0047] Another method within the scope of this invention, for which 20 operability is determined, is the step of having the resource transmit its operability status to the DNS server 66. In either method used, the DNS server 66 makes the resource available as long as there has been an operability status transmitted within the cycle time. If no transmission has occurred, then the resource is removed from the DNS server 66. If the resource begins to retransmit 25 its operability status, then the DNS server 66 returns it to the server, thus making it available to a request.

[0048] The many features and advantages of the invention are apparent from the detailed specification, and thus, it is intended by the appended claims to cover all such features and advantages of the invention, which fall within the true spirit and scope of the invention. Further, since numerous modifications and variations will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation illustrated and described, and accordingly, all suitable modifications and equivalents may be resorted to, falling within the scope of the invention.